We claim:

1. A cache memory, comprising:

a plurality of cache frames for storing information from main memory; and

an adaptive frame locking mechanism for locking a number of said most recently used frames.

2. The cache memory of claim 1, further comprising a memory for recording an identifier of the n most recently used frames.

3. The cache memory of claim 2, wherein said identifier is a frame address.

4. The cache memory of claim 2, wherein said identifier is a flag associated with said most recently used frames.

5. The cache memory of claim 2, wherein said identifier of the n most recently used frames is maintained for each of a plurality of tasks.

6. The cache memory of claim 1, wherein said adaptive frame locking mechanism does not lock all the frames in a set concurrently.

7. The cache memory of claim 1, wherein said number of said most recently used frames identifies the most recently accessed 3n/2 frames on average.

8. The cache memory of claim 1, wherein said adaptive frame locking mechanism includes three latches (a, b, and lock) for each frame of said cache.

9. The cache memory of claim 8, wherein said latch a is set when a frame is accessed and the value in latch a of a frame is transferred to latch b and latch a is reset after n accesses.

10. The cache memory of claim 8, wherein said adaptive frame locking mechanism sets a lock latch of a given frame, locking the frame, if either latch a or latch b is set when the lock signal is asserted.

11. The cache memory of claim 1, further comprising an adaptive frame unlocking mechanism that automatically unlocks frames that cause a performance degradation for a task.

12. The cache memory of claim 11, wherein said adaptive frame unlocking mechanism includes a counter for monitoring a number of times a task experiences a frame miss.

13. The cache memory of claim 1, wherein said cache is a two way set associative cache and said most recently used frames are identified by taking an inverse of a least recently used identifier.

14. The cache memory of claim 1, wherein said locking is performed if a first task is interrupted by a second task.

15. A method for locking frames in a cache memory, said method comprising the steps of:

storing information from main memory in frames of said cache memory;

monitoring a number of most recently used frames; and

locking said number of said most recently used frames if a task is interrupted by another task.

16. The method of claim 16, wherein said monitoring step maintains a frame address of said most recently used frames.

17. The method of claim 16, wherein said monitoring step maintains a flag associated with said most recently used frames.

18. The method of claim 16, wherein said monitoring step maintains an identifier of the n most recently used frames for each of a plurality of tasks.

19. The method of claim 15, wherein said locking step does not lock all the frames in a set concurrently.

20. The method of claim 15, wherein said number of said most recently used frames identifies the most recently accessed 3n/2 frames on average.

21. The method of claim 15, further comprising the step of automatically unlocking frames that cause a significant performance degradation for a task.

22. The method of claim 21, wherein said step of unlocking further comprises the step of monitoring a number of times a task experiences a frame miss.

23. A cache memory device comprising:

a memory element for storing information from main memory in frames of said cache memory;

means for monitoring a number of most recently used frames; and

means for locking said number of said most recently used frames if a task is interrupted by another task.

24. The cache memory of claim 23, wherein said means for locking said frames does not lock all the frames in a set concurrently.

25. The cache memory of claim 23, further comprising means for unlocking said locked frames that automatically unlocks frames that cause a significant performance degradation for a task.

26. The cache memory of claim 25, wherein said means for unlocking includes a counter for monitoring a number of times a task experiences a frame miss.

27. The cache memory of claim 23, wherein said cache is a two way set associative cache and said most recently used frames are identified by taking an inverse of a least recently used identifier.

28. The cache memory of claim 23, wherein said locking is performed if a first task is interrupted by a second task.

29. An integrated circuit, comprising:

a cache memory having a plurality of cache frames for storing information from main memory; and

an adaptive frame locking mechanism for locking a number of said most recently used frames.

30. The integrated circuit of claim 29, further comprising a memory for recording an identifier of the n most recently used frames.

31. The integrated circuit of claim 29, further comprising an adaptive frame unlocking mechanism that automatically unlocks frames that cause a performance degradation for a task.

32. The integrated circuit of claim 29, wherein said locking is performed if a first task is interrupted by a second task.